UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/600,284	06/20/2003	Steve Burns	50277-2139	9791

42425 7590 03/17/2009
HICKMAN PALERMO TRUONG & BECKER/ORACLE
2055 GATEWAY PLACE
SUITE 550
SAN JOSE, CA 95110-1083

EXAMINER	
TSUI, WILSON W	

ART UNIT	PAPER NUMBER
2178	

MAIL DATE	DELIVERY MODE
03/17/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/600,284	BURNS ET AL.	
	Examiner	**Art Unit**	
	WILSON TSUI	2178	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 04 December 2008.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-48 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-48 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. This Final Action is in response to the amendment filed on: 12/04/08.

2. Claims 1, and 18 are amended. Claims 47, and 48 are new. Claims 1-48 are pending.

3. The following rejections are withdrawn, as necessitated by applicant's amendment:

- Claims 1-3, 5, 7, 8, 13, 14, 16-26, 28, 30, 31, 36, 37, and 39-46 rejected under 35 U.S.C. 103(a) as being unpatentable over Abrams et al, in further view of Hofmann et al, and further in view of Griffen.
- Claims 4, 6, 27, and 29 rejected under 35 U.S.C. 103(a) as being unpatentable over Abrams et al, in view of Hofmann et al, in view of Griffen, and further in view of Hind et al.
- Claims 9 – 12, and 32 – 35 rejected under 35 U.S.C. 103(a) as being unpatentable over Abrams et al, in view of Hofmann et al, in view of Griffen, and further in view of and Hind et al, and further in view of Burnard et al.
- Claims 15 and 38 rejected under 35 U.S.C. 103(a) as being unpatentable over Abrams et al, Hofmann et al and in view of Griffen and further in view of Katariya et al.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claim 1-3, 5-14, 16-26, 28- 37, and 39-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Abrams et al (US Patent: 6,675,350 B1, issued: Jan. 6, 2004, filed: Nov. 4, 1999), in further view of Polizzi et al (US Application: US 2002/0052954 A1, published: May 2, 2002, filed: Apr. 27, 2001).

With regards to claim 1, Abrams et al teaches:

In response to receiving a request to display the page, performing the steps of:

- *Determining that the page is associated with a page parameter* (Fig 6: whereas a page is displayed according to user preference data/parameters for collecting data from a particular site/URL (column 4, lines 13-30))

- *Inspecting a mapping to determine that the page parameter is mapped to a portlet parameter* (As shown in Fig 2a, and Fig 2b, and explained in column 4, lines 13-30, the page parameters are checked and refined by a user (thus establishing mapping data with respect to a page parameter and a portlet parameter) to determine what summary information to display in each of the portlets (Fig 6, 630) *of a portlet that generates a component of the page that is based, at least in part, on the portlet parameter* (whereas, as explained in column 6, lines 12-25: each portlet receives parameters of the page, and each of the parameters are based upon user customized specified headlines of web sites).

- *Passing a value associated with the page parameter as a value of the portlet parameter to* a routine responsible for rendering the component, *and the* routine *generating the component based upon the value associated with the portlet parameter* (Fig 2a, and Fig 2b: whereas, the portlets use page parameters such as URL data to display page/summary information for the page located at the particular URL, and constraint based parameters to display constraint based page/summary information for the page located at the particular URL) *and inserting the component that was generated by the* routine *into the page* (Fig 6: whereas, the generated data/component is inserted into a page)

However, Abrams et al does not expressly teach

generating and storing a mapping that maps one or more page parameters to one or more portlet parameters, wherein the mapping is stored separate from pages associated with the one or more page parameters; ... determining that the page is associated with a page parameter *from the one or more page parameters*, *retrieving* … the mapping; and *wherein the portlet is executable code that is operable to generate page component;* passing a value associated with the page parameter as a value of the portlet parameter *to the portlet* that generates the component of the page, and inserting the component that was generated by the *portlet* into the page.

However, Polizzi et al teaches *generating and storing a mapping that maps one or more page parameters to one or more portlet parameters, wherein the mapping is stored separate from pages associated with the one or more page parameters; ...* determining that the page is associated with a page parameter *from the one or more page parameters*, *retrieving* … the mapping; and *wherein the portlet is executable code that is operable to generate page component;* passing a value associated with the page parameter as a value of the portlet parameter *to the portlet* that generates the component of the page, and inserting the component that was generated by the *portlet* into the page (paragraph 0030, 0032, 0033, 0038, and 0092: whereas, data is stored separate from the page itself, the data including metadata, which maps parameters to generate components of a page, by mapping one or more page parameters, such as user identification or preferences to appropriate portlets. Each portlet capable of generating a component of a page, such that the portal objects/components are appropriately retrieved, and assembled for an output portal page. Additionally, page parameters and metadata can be customized/generated, in order to instruct an appropriate portlet to generating a customized component of a page).

It would have been obvious to one of the ordinary skill in the art at the time of the invention to have modified Abrams et al's Portal aggregation method, such that a separately stored mapping is used to generate components of a page, using stored mapping parameters, as taught by Polizzi et al. The combination would have allowed Abrams et al to have "implemented a standardized, easy-to-learn method for retrieving

data through the use of an enterprise-wide computer system, which is connected to a variety of computer systems" (Polizzi et al, paragraph 0003).

With regards to claim 2, which depends on claim 1, Abrams et al, and Polizzi et al similarly teaches *wherein the step of mapping the page parameter, wherein mapping the page parameter comprises the steps of: mapping the page parameter to a second portlet parameter associated with a second component of the page; and passing the value associated with the page parameter as the value of the second portlet parameter to a second portlet that generates the second component*, as similarly explained in the rejection for claim 1, whereas multiple portlets receive one or more page parameters, and the URL data (value of the page parameter) is passed to logic/routine(s) responsible for rendering an updated/second page).

With regards to claim *3*, which depends on claim 1, Abrams et al, and Polizzi et al, similarly teach *wherein: establishing a plurality of page parameters for the page; mapping the plurality of page parameters to a plurality of portlet parameters associated with the component of web page; wherein the step of inspecting the mapping further comprises the step of inspecting the mapping to determine which page parameters of the plurality of page parameters are mapped to each of the plurality of portlet parameters; wherein the step of passing the value further comprises the step of passing, based on the mapping, values associated with the plurality of page parameters as the values of the plurality of portlet parameters to the portlet that generates; and*

wherein the step of the portlet generating the component further comprises the step of the portlet generating the component based upon the values associated with the plurality of portlet parameters, as similarly explained in the rejection for claim 1 (whereas multiple parameters are supported), and is rejected under similar rationale.

With regards to claim 5, which depends on claim 1, Abrams et al, and Polizzi et al similarly teaches *wherein the steps of mapping further comprises the step of mapping the page parameter to the portlet parameter and mapping a second page parameter to a second portlet parameter of the portlet that generates the component of the page*, as similarly explained in the rejection for claim 1, URL data is the first page parameter, and constraint based parameters are used as secondary parameters for the component of the page; and thus, rejected under similar rationale.

With regards to claim 6, Abrams et al teaches *establishing the page parameter, and passing the value associated with the page parameter further comprises the step of passing the value as the value of the portlet parameter the portlet that generates the component,* as similarly explained in the rejection for claim 1, and is rejected under similar rationale. However, Abrams et al does not expressly teach the value associated with the page parameter is a *default value*.

Yet, Polizzi et al teaches a *default value* (paragraph 0033: whereas, default values can be associated with page parameters, when retrieving the appropriate portlet).

It would have been obvious to one of the ordinary skill in the art at the time of the invention to have modified Abrams et al and Polizzi et al's method for mapping a page parameter to a portlet parameter, such that the page parameter value is a default value, as also taught by Polizzi et al. The combination of Abrams et al, Polizzi et al would have allowed Abrams et al to have "allowed a user to have been able to customize the content of his personal portal page by adding or removing certain portal objects or by modifying the content of certain portal objects" (Polizzi et al, paragraph 0030).

With regards to claim 7, which depends on claim 1, Abrams et al, and Polizzi et al similarly teach *wherein the request to display the page includes a URL and the URL includes the value associated with the page parameter, and wherein the step of passing the value associated with the page parameter is performed by passing the value contained in the URL as the value of the portlet parameter* (whereas, as explained in column 4, lines 13-30, and in the rejection for claim 1, URL data is used as parameter information, to be passed as the value of the portlet parameter).

With regards to claim 8, which depends on claim 1, Abrams et teaches *further comprising the steps of: presenting to a user a user interface for customizing the page; in response to the user interacting with the user interface, obtaining a user specified value for the page parameter; and wherein the step of passing the value associated with the page parameter is performed by passing the user specified value as the value of the portlet parameter to the portlet responsible for rendering the component*, (in column 4,

lines 1-12, and column 6, lines 25-32, a user interface is used by a user to specify page parameter values including URL, constraint, and layout/positions/fonts to a routine for rendering/displaying the component).

Additionally, as explained in the rejection for claim 1, the combination of Abrams et al and Polizzi et al teaches Abrams et al's routine is modified such that a portlet-method is used when *passing the value of the portlet parameter to the portlet that generates the component*.

With regards to claim 9, which depends on claim 1, Abrams et al teaches *determining a selected value based on override* preferences (column 6, lines 12-32: whereas, override settings/preferences are determined), and *passing the selected value as the value of the portlet parameter to the routine responsible for rendering the component* (as similarly explained in the claim 1, and also in column 6, lines 12-32, the selected preference values are used to render a customized view).

However Polizzi et al teaches a override *hierarchy,* and passing the user specified value as the value of the portlet parameter to the portlet that generates the component (paragraph 0033: whereas, a user specified value can override a default value of the portlet parameter to the portlet that generates the component).

It would have been obvious to one of the one of the ordinary skill in the art at the time of the invention to have modified. Abrams et al method for using portlets to assemble a page, such that one or more portlets can accept user specified parameters that override previous parameter values, as taught by Polizzi et al. The combination of Abrams et al

and Polizzi et al would have allowed Abrams et al to have have "implemented a standardized, easy-to-learn method for retrieving data through the use of an enterprise-wide computer system, which is connected to a variety of computer systems” (Polizzi et al, paragraph 0003).

With regards to claim 10, which depends on claim 9, Abrams et al teaches *the plurality of values includes a URL page parameter value* (as similarly explained in the rejection for claim 1) *and a customize page parameter value* (as similarly explained in the rejection for claim 1, whereas the constraint based parameters, are custom page parameter values), as well as *override* preferences (column 6, lines 12-32: whereas, override settings/preferences are determined). However, Abrams et al does not expressly teach an override *hierarchy that specifies that the URL page is the page parameter value is the selected value.*

Yet, the combination of Abrams et al and Polizzi et al teaches an override hierarchy, and a *default page parameter value as the selected value,* as similarly explained in the rejection for claim 9, and is rejected under similar rationale.

It would have been obvious to one of the ordinary skill in the art at the time of the invention to have modified Abrams et al's page parameter value, such as a URL; such that the override hierarchy is used to specify the value to be a selected value, as taught by Polizzi et al. The combination of Abrams et al and Polizzi et al would have allowed Abrams et al to have “allowed a user to customize content of his personal page, by

adding or removing certain portal objects, or by modifying the content of certain portal objects" (Polizzi et al, paragraph 0030).

With regards to claim 11, which depends on claim 9, Abrams et al teaches *the plurality of values includes a page parameter value* (as similarly explained in the rejection for claim 1) *and a customize page parameter value* (as similarly explained in the rejection for claim 1, whereas the constraint based parameters, are custom page parameter values), as well as *override* preferences (column 6, lines 12-32: whereas, override settings/preferences are determined). However, Abrams et al does not expressly teach an override *hierarchy that specifies that the customize page parameter value is the page parameter value is the selected value.*

Yet, the combination of Abrams et al, and Polizzi et al teaches *an override hierarchy that specifies that the customized page parameter value is the page parameter value is the selected value*, as similarly explained in the rejection for claim 9, and is rejected under similar rationale.

With regards to claim 12, which depends on claim 9, Abrams et al teaches *the plurality of values includes a page parameter value* (as similarly explained in the rejection for claim 1),a *portlet specified value* (as similarly explained in the rejection for claim 1), as well as *override* preferences (column 6, lines 12-32: whereas, override settings/preferences are determined). However, Abrams et al does not expressly teach,

the page parameter value is a *default value, and an* override *hierarchy that specifies that the default page parameter is the selected value.*

Yet, the combination of Abrams et al, and Polizzi et al teaches *an override hierarchy that specifies that the default page parameter is the selected value,* as similarly explained in the rejection for claim 9, and is rejected under similar rationale.

With regards to claim 13, which depends on claim 1, Abrams et al teaches *further comprising the step of presenting to a page designer a user interface for specifying the mapping between the page parameter and the portlet parameter* ((whereas, as explained in column 4, lines 1-12, and column 6, lines 25-32, a user interface is used by a user to specify page parameter values including URL, constraint, and layout/positions/fonts to a routine for rendering/displaying/mapping the component)).

With regards to claim 14, which depends on claim 1, Abrams et al, and Polizzi et al teaches registering the portlet with a portal repository, wherein the process of registering the portlet causes data associated with the portlet to be stored in the portal repository (Abrams et al, claim 1: whereas, a data source comprises registered profile data associated with the routine).

With regards to claim 16*,* which depends on claim 1*,* Abrams et al teaches *further comprising the step of receiving input from a page designer through a user interface to create the mapping between the portlet parameter and the page parameter* (whereas, as explained in column 4, lines 1-12, and column 6, lines 25-32, a user interface is used by a user to specify page parameter values including URL, constraint, and layout/positions/fonts to a routine for rendering/displaying/mapping the component)*.*

With regards to claim 17, which depends on claim 1, Abrams et al teaches the method further comprises the step of retrieving the stored value; and the step of the porltet generating the component further comprises the step of the portlet generating the component based upon the retrieved value (claim 1 of Abrams et al, Fig. 2A: whereas, the stored value(s)/preferences/constraints are stored in data stores, which are used to generate the components 240, 250, and 260). Additionally, as explained in the rejection for claim 1, the combination of Abrams et al and Polizzi et al teaches Abrams et al's routine is modified such that a portlet-method is used when passing the value of the portlet parameter to the portlet that generates the component.

With regards to claim *18.* Abrams et al teaches a method comprising:

- *In response to a user manipulating a component associated with a page,a portlet that generates the component generating a particular event* (column 4, lines 20-21: whereas a user manipulates a web address in component 220 of a portlet, causing the portlet in the page to generate a URL selection event)

*The portlet passing data that represents the particular event to logic associated with the page, inspecting a first mapping that maps events to actions and event output parameters to page parameters (*whereas, as explained in column 4, lines 1-12, and column 6, lines 25-32, a user uses a component/portlet (by generating an event as explained above) to specify page parameter values including URL, constraint, and layout/positions/fonts to a routine for rendering/displaying/mapping the component*), determining, based on the first mapping and the passed data, an action to perform in response to the particular event* (whereas the action to perform is to display all hyperlinks with their associated text for the selected site in pane 260 (column 4, lines 21-24))*; inspecting the first mapping to determine that an event output parameter associated with the particular event is mapped to a page parameter; and causing the action to be performed* … wherein *causing the action to be performed comprises passing a value of the event output parameter as the value of the page parameter* (column 4, lines 21-29: whereas, the URL data that represents the event is mapped to panes 240, 250, and 260, and a display action with regards to the URL data is performed).

However, Abrams et al does not expressly teach *generating and storing a first mapping that maps one or more events to one or more actions and one or more event output parameters to one or more page parameters*; a portlet *that previously generated* the component; *wherein the portlet is executable code that is operable to generate page components; and retrieving* the first mapping.

Yet Polizzi et al teach *generating and storing a first mapping that maps one or more*

events to one or more actions and one or more event output parameters to one or more page parameters (paragraph 0030, 0033, 0060: whereas, a mapping can be generated such that one or more time events are mapped to one or more actions/updates/scheduling of a job); a portlet *that previously generated* the component; *wherein the portlet is executable code that is operable to generate page components; and retrieving* the first mapping (paragraph 0030, 0060, 0092: whereas, portlets that previously generated components are retrieved and assembled, by retrieving mapping from metadata; to generate an updated page).

It would have been obvious to one of the ordinary skill in the art at the time of the invention to have modified Abrams et al's Portal aggregation method, such that events are mapped to one or more actions to generate components of a page, using stored mapping parameters, as taught by Polizzi et al. The combination would have allowed Abrams et al to have "implemented a standardized, easy-to-learn method for retrieving data through the use of an enterprise-wide computer system, which is connected to a variety of computer systems" (Polizzi et al, paragraph 0003).

With regards to claim 19, which depends on claim 18, Abrams et al teaches *wherein the page is a first page and the page parameter is associated with a second page; and the step of causing the action to be performed further comprises the step of passing the value of the page parameter to logic responsible for rendering a second page*, as similarly explained in claim 1, the URL data (value of the page parameter) is passed to logic/routine(s) responsible for rendering an updated/second page, and is rejected under the similar rationale.

With regards to claim 20, which depends on claim 18, Abrams et al teaches *wherein the step of causing the action to be performed further comprises the step of generating a request that specifies a URL, wherein the value of the page parameter is included in the URL*: (whereas, as explained in column 4, lines 13-30, and in the rejection for claim 1, URL data is used as parameter information, to be passed as the value of the portlet parameter).

With regards to claim 21, which depends on claim 20, Abrams et al teaches*: the step of generating the request further comprises the step of generating a request for executable code; and the step of causing the action to be performed further comprises the step of invoking the executable code*, as similarly explained in the rejection for claim 1, page parameter data is passed to the appropriate portlet parameters, and the passing of the value causes the display/render action to be performed. Since the rendering as shown in Fig 2A as performed/executed, the figure inherently teaches that code must have been executed in order for the appropriate components/portlets to have been updated with the mapped parameter values.

With regards to claim 22, which depends on claim 21, Abrams et al teaches *wherein the executable code is a web service* (column 1, lines 45-60: whereas, the executable code, provides user's with a service to collect information from disparate sources, to be displayed in a summarized and consistent manner).

With regards to claim *23*, which depends on claim 18, Abrams et al teaches wherein*:*

The action comprises rendering a second page, wherein the page parameter is associated with the second page, and wherein rendering the second page (as similarly explained in the rejection for claim 19, and is rejected under similar rationale) comprises the steps of:

Inspecting a mapping to determine that the page parameter is mapped to a portlet, as similarly explained in the rejection for claim 1, and is rejected under similar rationale.

The second portlet generating a component based upon the value associated with the portlet parameter, as similarly explained in the rejection for claim 19, and is rejected under similar rationale.

Inserting the second component that was generated by a portlet into a page, as similarly explained in the rejection for claim 19, and is rejected under similar rationale.

However, Abrams et al does not expressly teach Inspecting a *second mapping* to determine that the page parameter is mapped to a portlet parameter of a *second* portlet that generates a *second* component of the *second* page that is based, at least in part, on the portlet parameter; and Passing the value of the page parameter as the value of the portlet parameter to the *second* portlet.

Yet, the combination of Abrams et al and Polizzi et al similarly teaches Inspecting a *second mapping* to determine that the page parameter is mapped to a portlet parameter of a *second* portlet that generates a *second* component of the *second* page that is based, at least in part, on the portlet parameter; and Passing the value of the page parameter as the value of the portlet parameter to the *second* portlet, as explained in

the rejection for claim 1 (and also as explained in paragraph 0092 of Polizzi et al: whereas, more than one portlet is retrieved to generate a page comprising a plurality of components);

With regards to claim 24, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 1, and is rejected under similar rationale.

With regards to claim 25, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in claim 2, and is rejected under similar rationale.

With regards to claim 26, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 3, and is rejected under similar rationale.

With regards to claim 28, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 5, and is rejected under similar rationale.

With regards to claim 29, for a computer-readable storage medium storing one or

more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 6, and is rejected under similar rationale.

With regards to claim 30, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 7, and is rejected under similar rationale.

With regards to claim 31, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 8, and is rejected under similar rationale.

With regards to claim 32, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 9, is rejected under similar rationale.

With regards to claim 33, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 10, is rejected under similar rationale.

With regards to claim 34, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in claim 11, is rejected under similar rationale.

With regards to claim 35, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 12, and is rejected under similar rationale.

With regards to claim 36, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 13, and is rejected under similar rationale.

With regards to claim 37, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 14, and is rejected under similar rationale.

With regards to claim 39, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the m the method recited in claim 16, and is rejected under similar rationale.

With regards to claim 40, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 17, and is rejected under similar rationale.

With regards to claim 41, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 18, and is rejected under similar rationale.

With regards to claim 42, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 19, and is rejected under similar rationale.

With regards to claim 43, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 20, and is rejected under similar rationale.

With regards to claim 44, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 21, and is rejected under similar rationale.

With regards to claim 45, for a teaches a computer-readable storage medium

storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 22, and is rejected under similar rationale.

With regards to claim 46, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 23, and is rejected under similar rationale.

With regards to claim 47, which depends on claim 1, the combination of Abrams and Polizzi et al teaches *wherein the portlet is a first portlet and wherein the mapping maps a single page parameter, of the one or more page parameters, top a first portlet parameter of the first portlet, and to a second portlet parameter of a second portlet,* as similarly explained in the rejection for claim 1 (whereas, Polizzi et al teaches that portlets can be customized/accept parameters, and that one or more portlets can be selected to render components of a page), and is rejected under similar rationale.

With regards to claim 48, which depends on claim 1, the combination of Abrams and Polizzi et al teaches *a computer readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in claim 47,* as similarly explained in the rejection for claim 47, and is rejected under similar rationale.

5. Claims 4, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Abrams et al (US Patent: 6,675,350 B1, issued: Jan. 6, 2004, filed: Nov. 4, 1999),in view of Polizzi et al (US Application: US 2002/0052954 A1, published: May 2, 2002, filed: Apr. 27, 2001), and further in view of Hind et al (US Application: 2004/0205555 A1, published: Oct. 14, 2004, filed: Sep. 18, 2001)

With regards to claim 4, which depends on claim 1, Abrams et al teaches *mapping the page parameter to the portlet parameter associated with the component of the page*, as similarly explained in the rejection for claim 1, and is rejected under similar rationale. However, Abrams et al does not expressly teach … *without mapping the page parameter to portlet parameters associated with any other components of the page*. Hind et al teaches … *without mapping the parameters to portlet parameters associated with any other components of the page* (Fig. 3A, paragraph 0024: whereas, content for some components/portlets are updated, while some are not, and thus components/portals are selectively mapped for receiving parameter data).

It would have been obvious to one of the ordinary skill in the art at the time of the invention to have modified Abrams et al and Hofmann et al's method for mapping page parameters, to have only mapped page parameters to a select page component, as taught by Hind et al. The combination of Abrams et al, Polizzi et al, and Hind et al would have allowed Abrams et al to have "reduced the time a user waits for receiving a portal page [by] spawning individual threads for reach portlet" (Hind et al, paragraph 0009).

With regards to claim 27, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 4, and is rejected under similar rationale.

6. Claims 15 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Abrams et al (US Patent: 6,675,350 B1, issued: Jan. 6, 2004, filed: Nov. 4, 1999), in view of in view of Polizzi et al (US Application: US 2002/0052954 A1, published: May 2, 2002, filed: Apr. 27, 2001), and further in view of Katariya et al (US Patent: 6,564,251 B2, issued: May 13, 2003, filed: Dec. 3, 1998).

With regards to claim 15, which depends on claim 14, Abrams et al ,and Polizzi et al teaches *the data associated with the portlet,* and *communicated with the portal repository*, as similarly explained in the rejection for claim 14, and is rejected under similar rationale. However, Abrams et al, and Polizzi et al do not expressly teach the data associated with the portlet, is communicated to the portal repository *as an XML document*.

Katariya et al teaches communicating with the portal repository, through the use of an *XML document* (columns 5 and 6, lines 59-67 and 1-9 respectively: whereas, preference/parameter information is communicated to a portal repository via XML format).

It would have been obvious to one of the ordinary skill in the art at the time of the invention to have modified Abrams et al's communication of data to a portal repository, such that the data is passed using a XML document, as similarly taught by Katariya et al. The combination of Abrams et al, Polizzi et al, and Katariya et al would have allowed Abrams et al to have allowed "the content of each page to have been enhanced by the rendered data from the provider objects, thereby adding dynamic behavior to the predefined page" (Katariya et al, column 2, lines 26-31).

With regards to claim 38, for a computer-readable storage medium storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform a method similar to the method recited in claim 15, and is rejected under similar rationale.

Response to Arguments

7. Applicant's arguments with respect to claims 1-48 have been considered but are moot in view of the new ground(s) of rejection.

8. With respect to claim 1, the applicant argues that the GUI display areas of Abrams Figs. 2A, 2B, and 6 are not pages, but rather GUI display areas, generated by an application tool that provides an HTML parser (ABRAMS, col. 3, lines 48-53).

9. However, this argument is not persuasive since, the applicant is arguing that the pages must be a particular type of page, such as an HTML coded page. Yet, as claimed within claim 1, there is no requirement as to the extent of a particular type of page being

required; only that a page is required. Thus, since Abrams teaches displaying a page of information (Fig 6, column 6, lines 12-25), then the examiner respectfully points out that Abrams still teaches the claimed 'page limitation'. Additionally, with regards to amended ‘mapping that is stored separate', from the pages, the Examiner respectfully directs the applicant to the new combination of references for the rejection of claim 1 above, which uses Polizzi et al, in combination with Abrams, to further show portal page generation using a mapping separate from the pages associated with one or more page parameters.

10. Additionally, with respect to claim 1, the applicant argues that Abrams does not describe or suggest that a mapping between page parameters and portlet parameters is inspected during the processing of the user input by the application tool.

11. However, this argument is not persuasive since Abrams teaches, *using* a user defined display structure (column 6, lines 12-25), *to generate a* customized portlet/panel in a portal page (as shown in Fig. 6, and explained column 6, lines 12-25). Thus, a correlation between user defined data/preferences of a page is identified with respect to how data is generated for a component of a page; and therefore the examiner respectfully points out that a mapping *is* implemented, for proper generation of a customized portal page (Fig. 6).

12. With respect to claim 18, the applicant argues (in page 19 of applicant remarks) that “even if one is to consider that the GUI and the panes may somehow correspond to a page, ... the GUI and any input provided by the user into such GUI would be processed internally within the execution tool”, … [and] "in contrast, claim 18 features a

first mapping that is stored separate from pages associated with the one or more page parameters". However, this argument is not persuasive, and the examiner respectfully directs the applicant to the explanation to the response to remarks for claim 1, above, as to how Abrams still teaches the claimed 'page', and also to the rejection for claim 18 above, which explains how Polizzi et al is used in combination with Abrams to teach the newly amended limitation.

13. Also with respect to claim 18, the applicant argues (page 20 of applicant's remarks), that "Abrams describes that when a user selects a web address in pane 220 of the GUI, the application tool displays in pane 260 all hyperlinks of the site indicated by the web address. This, however, does not describe or even suggest that any determination is made based on a mapping that maps one or more events to one or more actions and one or more event output parameters to one or more page parameters."

14. However, this argument is not persuasive. Since, a selection event is taught in Abrams, which *results identifying the selection* (column 4, lines 1-12, and column 6, lines 12-32), and furthermore, the identified selection (event output) is used as parameters to generate a portal page (shown in Fig 6) *such that* the selection customized selection(s) (column 6, lines 12-24) are used. Thus, Abrams teaches the mapping limitation as claimed.

15. With respect to depending claims 2-17,and 19-48 being allowable, since they depend upon an independent claim that is allowable; is not persuasive since the independent claims have been shown/explained to be rejected.

Conclusion

16. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to WILSON TSUI whose telephone number is (571)272-7596. The examiner can normally be reached on Monday - Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen Hong can be reached on (571) 272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/CESAR B PAULA/
Primary Examiner, Art Unit 2178

/Wilson Tsui/
Patent Examiner
Art Unit: 2178
March 08, 2009